# 北京网迅科技有限公司

# 千兆和万兆网卡 SR-IOV 驱动用户手册

2.0.0 版


**北京网迅科技有限公司**

**2024 年 01 月**

# 历史

| 版本 | 描述 | 发布/日期 |
| --- | --- | --- |
| 1.0.0 | 1、介绍 SR-IOV 基本操作。 | FengjuZhang<br>*2019 年 12 月 30 日* |
| 1.1.0 | 1、介绍 SR-IOV 直通到虚拟机下;<br>2、介绍 SR-IOV 创建失败 | ShuhuiWang<br>*2021 年 03 月 09 日* |
| 1.2.0 | 1、完善测试步骤<br>2、添加注意事项<br>3、添加异常解决方法 | FengjuZhang<br>*2021 年 08 月 23 日* |
| 2.0.0 | 1、添加千兆 ngbevf 使用方法<br>2、修改编译方法及部分细节 | FengjuZhang<br>*2024 年 01 月 17 日* |

# 目录

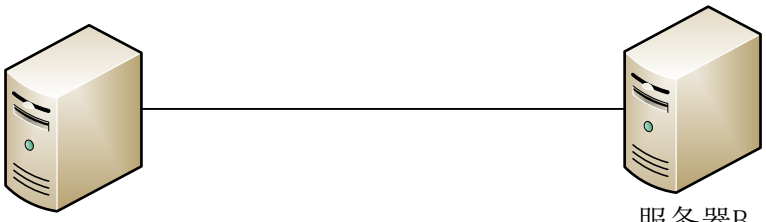
# 一、须知

在使用网迅千兆 ngbevf 和万兆 txgbevf 驱动前，建议仔细阅读本手册的全部内容。因为本手册对与千兆网卡虚拟化驱动（ ngbevf ）和万兆网卡虚拟化驱动 （ txgbe vf ）的使用方法、使用的流程，甚至使用中可能遇到的问题均有描述，可以帮助用户更快了解网迅 VF 驱动的使用方法。如果对手册内容存疑或者在阅读完手册后仍有其他问题，请及时联系网迅公司的销售或者技术支持人员，谢谢 ！

## 1. 准备

在网迅千兆和万兆网卡上使用 SR-IOV 功能前，请确保以下准备工作正确完成：

a．千兆或万兆网卡安装在服务器 A 和服务器 B 上，两者使用网线或光纤直连，测试拓扑如下图




b．服务器 A 和服务器 B 上，

千兆网卡：网迅物理网卡 ngbe 驱动加载完成（编译加载过程参考《网迅千兆网卡驱动使用方法》）。

万兆网卡：网迅物理网卡 txgbe 驱动加载完成（编译加载过程参考《网迅万兆网卡驱动使用方法》）。

c．服务器 A 和服务器 B 网口处于 LinkUp 状态，并可 ping 通

## 2. 驱动解压&编译

a． 解压

千兆：unzip ngbevf.zip

万兆：unzip txgbevf.zip

b． 切到源码目录

千兆：cd ngbevf/src

万兆：cd txgbevf/src

c． 编译（千兆 ngbevf 和万兆 txgbevf 编译命令相同）

注：编译时源码路径中含有中文、特殊字符、空格、标点符号，可能会导致编译失败
make modules_install只会安装驱动模块本身
make install会安装模块，并更新initramfs
不同平台上编译，安装的命令不同：

- **KylinV10/银河麒麟操作系统：**

编译：
make CHNOS=KYLIN

安装：
make CHNOS=KYLIN modules_install
或者
make CHNOS=KYLIN install

- **UOS 操作系统：**

编译：
make CHNOS=UOS
安装
make CHNOS=UOS modules_install
或者
make CHNOS=UOS install

- **Euler操作系统：**

编译：
make CHNOS=EULER
安装：
make CHNOS=EULER modules_install
或者
make CHNOS=EULER install

- **其他平台操作系统：**

编译：
make
安装：
make modules_install
或者
make install

### 3. 加载&卸载

步骤 2 编译之后，可进行驱动加载和卸载。

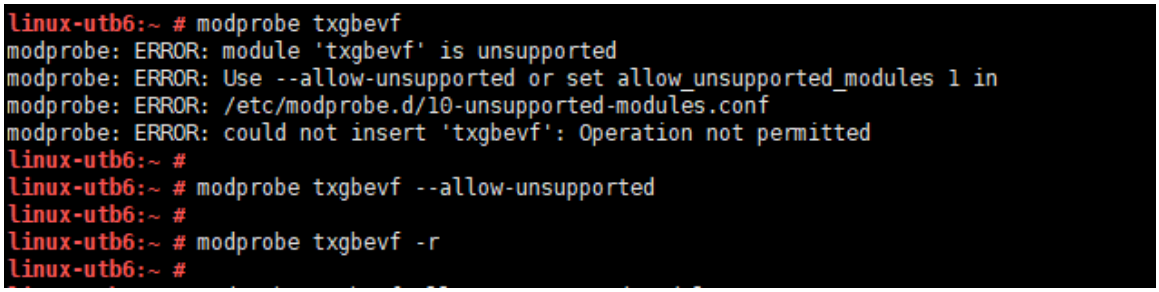
加载：modprobe txgbevf　　#需在编译完成之后

卸载：modprobe txgbevf　-r　　#需在驱动加载之后

注：在 suse 虚拟机下，加载驱动若报 ERROR（如下图），需在加载驱动时添加参数，才可加载成功：

千兆：modprobe ngbevf --allow-unsupported

万兆：modprobe txgbevf --allow-unsupported



## 4. 创建 SR-IOV

前提：需要将物理机 A 对应的 PF 口 up 起来，如 ifconfig ethA up
如创建 2 个 SR-IOV：echo 2 > /sys/class/net/ethA/device/sriov_numvfs
lspci -d 8088: //通过 pcie 查看虚拟网口

## 5. 关闭 SR-IOV

需要先将 VF 驱动卸载掉，然后关闭 SR-IOV:

**千兆：**

modprobe ngbevf -r
echo 0 > /sys/class/net/ethA/device/sriov_numvfs

注1. 当需要卸载 PF ngbe 驱动时，也需要先卸载 ngbevf 驱动，关闭 SR-IOV，最后才可以卸载 ngbe 驱动；

注2. 如需要 down ngbevf 虚拟网口，需要先 down 掉 ngbe 物理网口。

**万兆：**

modprobe txgbevf -r
echo 0 > /sys/class/net/ethA/device/sriov_numvfs

注3. 当需要卸载 PF txgbe 驱动时，也需要先卸载 txgbevf 驱动，关闭 SR-IOV，最后才可以卸载 txgbe 驱动；

注4. 如需要 down txgbevf 虚拟网口，需要先 down 掉 txgbe 物理网口。

## 6. 更改 SR-IOV 数量

若物理机上已经创建了 VF，但需要更改 VF 网口的数量，须先卸载 txgbevf 驱动，然后关闭 SR-IOV。如从 2 个 VF 网口改为 4 个 VF 网口:

千兆:

modprobe ngbevf -r
echo 0 > /sys/class/net/ethA/device/sriov_numvf
echo 4 > /sys/class/net/ethA/device/sriov_numvf

万兆:

modprobe txgbevf -r
echo 0 > /sys/class/net/ethA/device/sriov_numvf
echo 4 > /sys/class/net/ethA/device/sriov_numvf

## 7. 虚拟网口配置

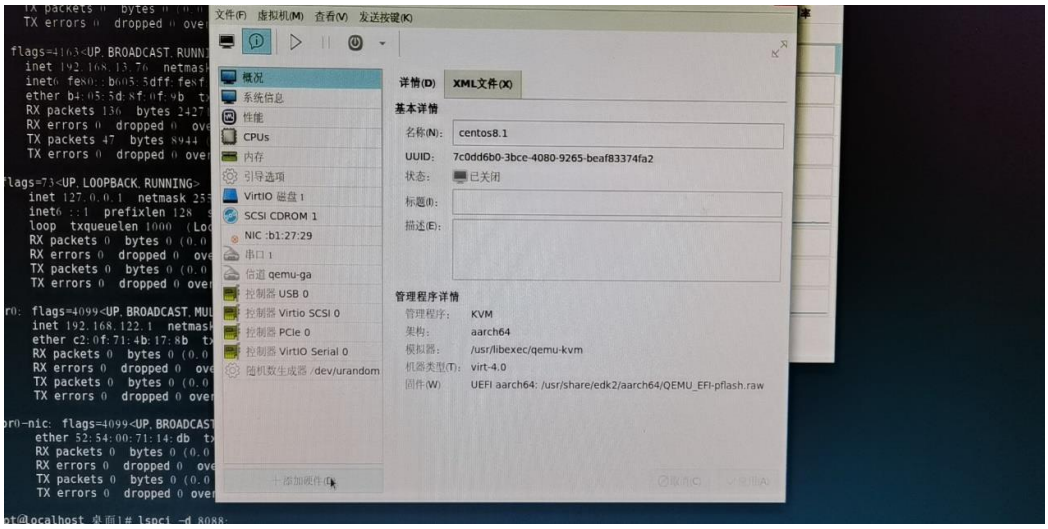
虚拟网口（VF）主要有以下两种使用场景，请根据实际情况，挑选场景进行测试。

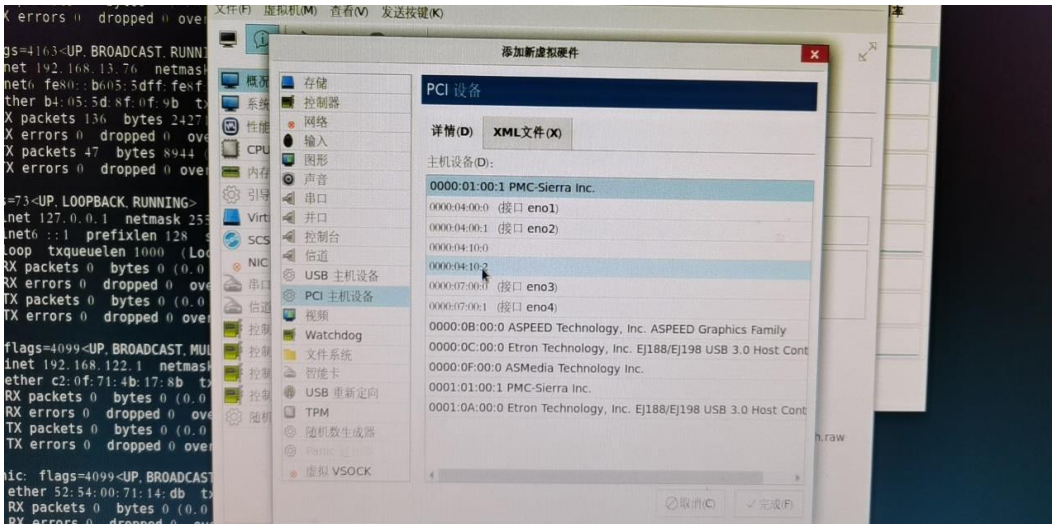
## 7.1 将虚拟网口（VF）配置在物理机下

a． 在服务器 A 上，将对应的物理网口 up 起来，如 ifconfig eth0 10.10.10.10/24 up
b． 在服务器 A 物理机上，先卸载原有的虚拟化 vf 驱动（若没有加载 vf 驱动，可忽略此步）：千兆：rmmod ngbevf　万兆：rmmod txgbevf
c． 示例：在服务器上创建 2 个 VF 网口(千兆实际可创建虚拟网口数量最大为 7 个，万兆实际可创建虚拟网口数量最大为 63 个)，其中 ethA 为服务器 A 上的物理网口名
echo 0 > /sys/class/net/ethA/device/sriov_numvfs //清空原有虚拟网口
echo 2 > /sys/class/net/ethA/device/sriov_numvfs //创建 2 个虚拟网口
d． 使用对应命令编译安装虚拟化驱动（命令请参考第 2 章 驱动解压&编译）
e． 加载虚拟网卡驱动
千兆：modprobe ngbevf　//加载千兆虚拟网卡驱动
万兆：modprobe txgbevf　//加载万兆虚拟网卡驱动
f． ip link show | grep "vf" |wc -l　//可查看到步骤 a 创建 VF 的个数
lspci -d 8088:　　//通过 pcie 查看虚拟网口
g． 在服务器 A 端配置虚拟网口 0 的 IP:
ifconfig ethA_0 192.168.11.10/24 up（注意：不可与 PF 同一网段）
在服务器 B 配置 ethB 的 IP:
ifconfig ethB 192.168.11.11/24 up
d． 使用 ping 命令检查网络连通性。

## 7.2 将虚拟网口（VF）直通到虚拟机（VM）下

a． 在服务器 A 上，将对应的物理网口 up 起来，如 ifconfig eth0 10.10.10.10/24 up
b． 在服务器 A 物理机上，在服务器 A 物理机上，先卸载原有的虚拟化 vf 驱动（若没有加载 vf 驱动，可忽略此步）：千兆：rmmod ngbevf　万兆：rmmod txgbevf
c． 示例：在服务器上创建 2 个 VF 网口(千兆实际可创建虚拟网口数量最大为 7 个，万兆实际可创建虚拟网口数量最大为 63 个)，其中 ethA 为服务器 A 上的物理网口名
echo 0 > /sys/class/net/ethA/device/sriov_numvfs //清空原有虚拟网口
echo 2 > /sys/class/net/ethA/device/sriov_numvfs //创建 2 个虚拟网口
d． 使用对应命令编译安装虚拟化驱动（命令请参考第 2 章 驱动解压&编译）
e． lspci -d 8088:　　//通过 pcie 查看虚拟网口
在虚拟机 vm 中将虚拟网口直通到虚拟机上：左下角 > 添加硬件 > pcie 设备




f． 启动虚拟机，在虚拟机上编译加载虚拟化 vf 驱动，详见第 2 章和第 3 章（若启动虚拟机报错，请参考第 11 章内容或联系服务器厂商确认该机器是否支持 SR-IOV 功能）

千兆：lsmod | grep ngbevf　　//查看是否安装了 ngbevf 驱动

万兆：lsmod | grep txgbevf　//查看是否安装了 txgbevf 驱动

ifconfig -a //查看虚拟网口是否加载成功

g． 在服务器 A 端配置虚拟网口 0 的 IP:

ifconfig ethA_0 192.168.11.10/24 up

在服务器 B 配置 ethB 的 IP:

ifconfig ethB 192.168.11.11/24 up

h． 使用 ping 命令检查网络连通性

**注：lspci 直通到虚拟机后，物理机 lspci 仍可看到 VF pcie 号，且虚拟机下 pcie 和物理机下不一样。**

## 9. 个性化命令

加载完 VF 驱动后，可以通过一些命令执行个性化操作。比如修改 MAC、mtu，测试性能等。

### 9.1 修改 VF 的 MAC 地址

ip link set dev ethA vf 0 mac 00:16:3e:67:75:10

在修改 VF 的 MAC 之后需要重新卸载&加载 VF 驱动后，新 MAC 方可生效。

### 9.2 修改 VF 的 mtu 地址

PF 和 VF 的默认 mtu 为 1500，最大值为 9414。在修改 VF mtu 时，需 VF 的 mtu 不大于 PF 的 mtu。若需将 VF 的 mtu 修改为大于 1500，需先将 PF 的 mtu 修改为大于等于 1500。

例：PF 和 VF 的 mtu 为 1500，需将 VF 的 mtu 修改为 2000，步骤如下：

a． ifconfig ethA mtu 9414　　#修改 PF 的 mtu 不小于 2000

b． ifconfig ethA_0 mtu 2000　#修改 VF 的 mtu 为 2000

## 10. NameSpace

### 10.1 NameSpace 的创建

当需创建多个 VF 时，可使用 NameSpace 进行使用及测试（配置服务器 B 的 PF 网口 IP 为 192.168.10.11）。

a． 将 ethA_0 绑定在 TestNS0 虚拟网络环境

ip netns add TestNS0　　#添加虚拟网络命名空间 TestNS0

ip link set dev ethA_0 netns TestNS0　　#将 ethA_0 添加到 TestNS0 虚拟网络环境

ip netns exec TestNS0 ifconfig ethA_0 192.168.10.100/24 up　# 配置 TestNS0 的 IP

b． 将 ethA_1 绑定在 TestNS1 虚拟网络环境

ip netns add TestNS1　　#添加虚拟网络命名空间 TestNS1

ip link set dev ethA_1 netns TestNS1　　#将 ethA_1 添加到 TestNS1 虚拟网络环境

ip netns exec TestNS1 ifconfig ethA_1 192.168.10.101/24 up　# 配置 TestNS1 的 IP

### 10.2 NameSpace 的使用

创建 NameSpacce 后，可在对应的 NameSpace 环境里执行各种操作，如：

ip netns exec TestNS0 arping –I ethA_0 192.168.10.10　　# TestNS0 本端 ping 对端

ip netns exec TestNS0 arping -I ethA_0 192.168.10.101　　# TestNS0 ping TestNS1

### 10.3 进入&退出 NameSpace

ip netns exec TestNS0 bash　　#进入 NameSpace

exit　　#退出 NameSpace

## 11. 异常

### 11.1 创建 SR-IOV 失败

当您在使用过程中出现虚拟网口（VF）无法创建的情况，请先检查 bios 里是否开启 SR-IOV 选项（即 VT-D 参数），若已开启，则需要在 os 下修改 grub 启动项：

1) vim /etc/default/grub
在 GRUB_CMDLINE_LINUX 后面加上'iommu=pt intel_iommu=on pci=realloc'

2) grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg #其中 centos 需要改成当前环境值
在/etc 路径下，你能看到：
[root@localhost etc]# ll | grep boot
lrwxrwxrwx. 1 root root 30 May 5 23:31 extlinux.conf
-> ../boot/extlinux/extlinux.conf
lrwxrwxrwx. 1 root root 22 May 5 23:32 grub2.cfg -> ../boot/grub2/grub.cfg
lrwxrwxrwx. 1 root root 31 May 5 23:36 grub2-efi.cfg
-> ../boot/efi/EFI/centos/grub.cfg

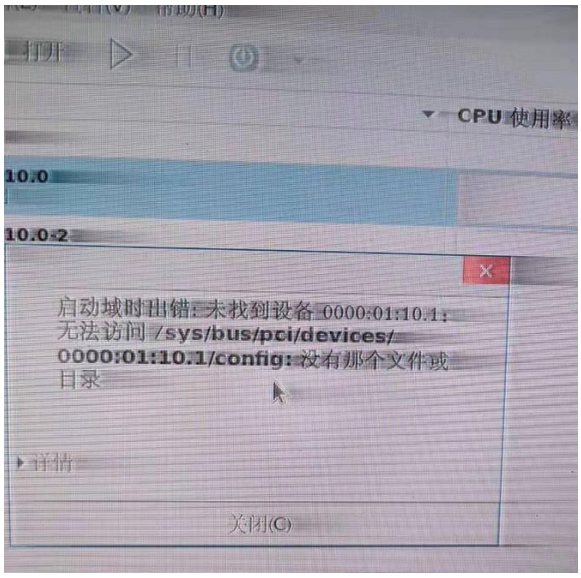
3) 重启机器

4）检查添加的启动项是否生效：
# more /proc/cmdline
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-957.el7.x86_64 root=/dev/mapper/centos-root ro crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet iomem=relaxed iommu=pt intel_iommu=on pci=realloc

### 11.2 启动虚拟机失败

提示启动域时出错：未找到设备 0000:01:10.1：无法访问 /sys/bus/pci/devices/0000:01:10.1/config:没有那个文件会目录，如下图：

此报错是因为物理机没有创建此 pcie 号（0000:01:10.1），可在"添加新虚拟硬件"界面删除此 pcie 号或在物理机下创建 SRIOV（会自动创建虚拟 pcie 号），可通过 lspci -d 8088: 查看 pcie 号。


CPU 使用率
10.0
10.0.2
启动域时出错：未找到设备 0000:01:10.1：
无法访问 /sys/bus/pci/devices/
0000:01:10.1/config：没有那个文件或
目录
详情
关闭(C)